



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Κεφάλαιο 9

Επιμέλεια:

Βασίλης Παλιουράς , Αναπληρωτής Καθηγητής
Ευάγγελος Δερματάς , Αναπληρωτής Καθηγητής
Σταύρος Νούσιος , Βοηθός Ερευνητή

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου των διδασκόντων καθηγητών.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών» έχει χρηματοδοτηθεί μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ανάπτυξη

Το παρόν εκπαιδευτικό υλικό αναπτύχθηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

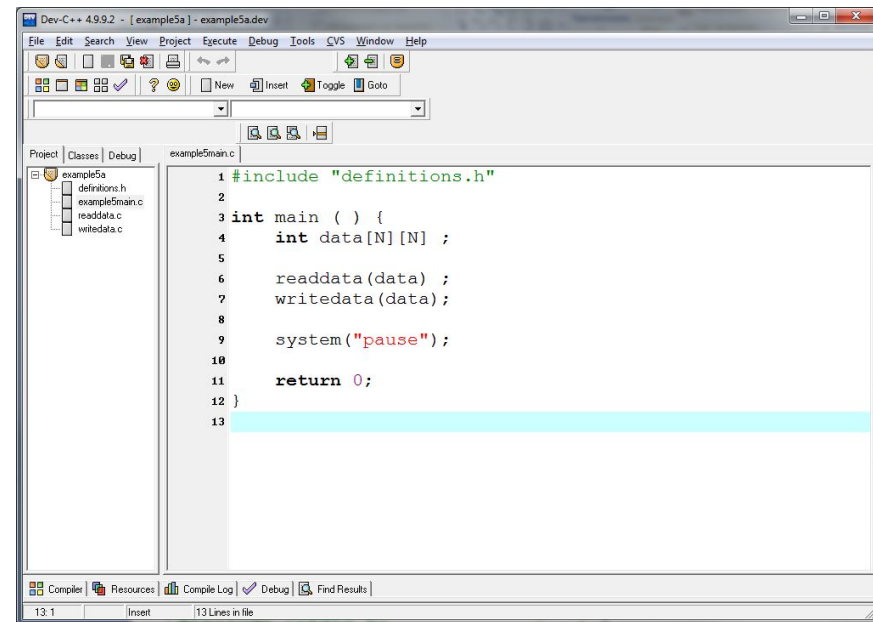
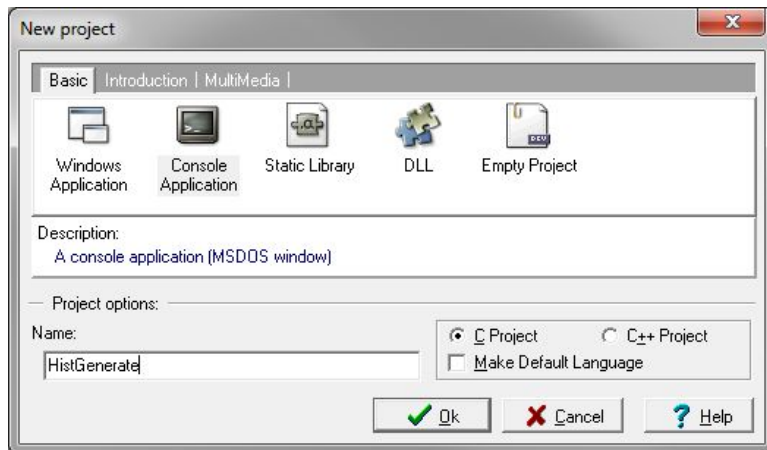
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Project σε Dev-C++



```
#include <stdio.h>
#define N 3

int main ( ) {
int i, j;

int a[N][N] = {{1,2,3}, {3,2,1}, {1,1,1}};
int *b = &a[0][0];

for (i=0; i< N; i++) {
    for (j=0; j< N; j++)
        printf("%d ",a[i][j]);
    printf("\n");
}

for (i=0; i< N*N; i++)
    printf("%d ", *(b+i));

system("pause");
return 0;
}
```



Συνάρτηση της βασικής βιβλιοθήκης scanf ()

```
int number;
```

```
char ch;
```

%d θα διαβάσει ακέραιο

```
scanf("%d", &number);
```

%c θα διαβάσει χαρακτήρα

```
scanf("%c", &ch);
```

τελεστής διεύθυνσης (&):

Επιστρέφει τη διεύθυνση

της θέσης μνήμης η οποία

αντιστοιχεί στη μεταβλητή

που ακολουθεί



Οργάνωση σε πολλά αρχεία (2)

```
#include "definitions.h"
```

```
int main ( ) {  
    int data[N][N] ;  
  
    readdata(data) ;  
    writedata(data);  
  
    system("pause");  
    return 0;  
}
```

```
#include "definitions.h"
```

```
void readdata(int a[N][N]) {  
    int i,j;  
    for (i =0 ; i < N ; i++)  
        for ( j = 0 ; j < N ; j ++){  
            printf ("element (%d,%d)?\t", i, j);  
            scanf("%d", &a[i][j]);  
        }  
}
```

```
#include "definitions.h"
```

```
void writedata(int b[N][N]) {  
    int i,j;  
    for (i =0 ; i < N ; i++) {  
        for ( j = 0 ; j < N ; j ++)  
            printf ("%d\t", b[i][j]);  
        printf("\n");  
    }  
}
```

To αρχείο definitions.h:

```
#define N 2  
#include <stdio.h>  
void readdata(int [N][N]);  
void writedata(int [N][N]);
```



Παράδειγμα

- Ζητήθηκε από 40 φοιτητές να βαθμολογήσουν το φαγητό στο κυλικείο από 1 (απαίσιο) έως και 10 (αστέρι michelin).
- Να συνοψίσουμε τα αποτελέσματα.
- Από Deitel & Deitel.



Παράδειγμα

- Λεκτική περιγραφή – Προστακτικός προγραμματισμός:
(οργάνωση προγράμματος – συναρτήσεις)

Διάβασε τις απαντήσεις

Υπολόγισε το ιστόγραμμα => `generatehist()`

Παρουσίασε το αποτέλεσμα => `printhist()`

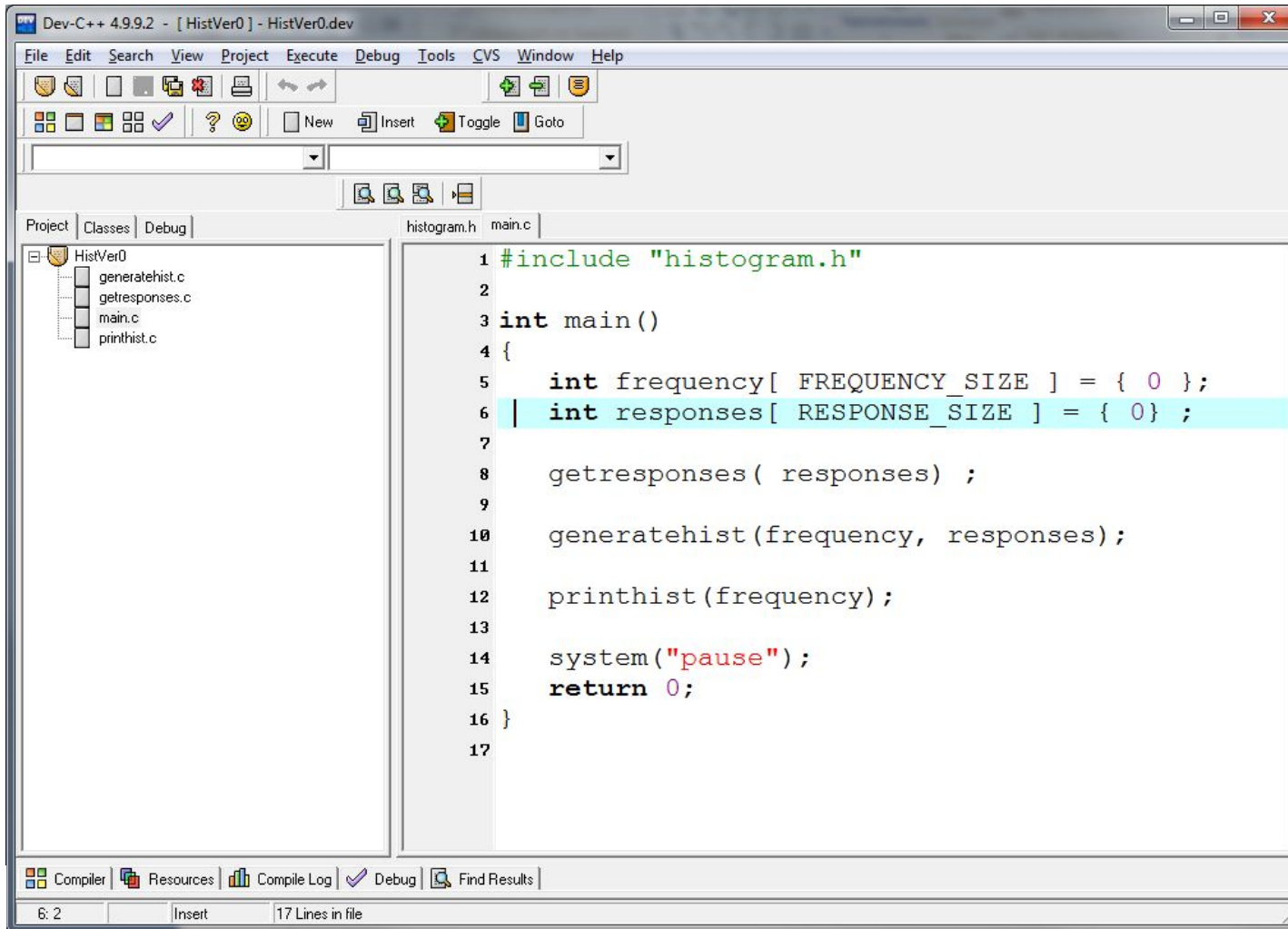
- Αναπαράσταση δεδομένων

– Πίνακας για απαντήσεις ↴ `responses[]`

– Πίνακας για ιστόγραμμα ↴ `frequency[]`



Version 0: οργάνωση προγράμματος



The screenshot shows the Dev-C++ 4.9.9.2 IDE interface. The title bar reads "Dev-C++ 4.9.9.2 - [HistVer0] - HistVer0.dev". The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations and development. The Project Explorer on the left shows a project named "HistVer0" with the following files: generatehist.c, getresponses.c, main.c, and printhist.c. The main editor window displays the code for main.c, which includes the following C code:

```
1 #include "histogram.h"
2
3 int main()
4 {
5     int frequency[ FREQUENCY_SIZE ] = { 0 };
6     int responses[ RESPONSE_SIZE ] = { 0 } ;
7
8     getresponses( responses ) ;
9
10    generatehist( frequency, responses ) ;
11
12    printhist( frequency ) ;
13
14    system( "pause" ) ;
15    return 0 ;
16 }
17
```

The status bar at the bottom indicates "6: 2", "Insert", and "17 Lines in file".



Version 0 - συναρτήσεις

```
1 #include "histogram.h"
2
3 void getresponses(int data[]) {
4     printf("getresponses\n");
5 }
6
```



Version 1: ελέγχουμε τις διεπαφές

```
E:\paliuras\courses\Principles\1213\lecture08\histo\ve
getresponses
generatehist
printhist
0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
Press any key to continue . . . _
```

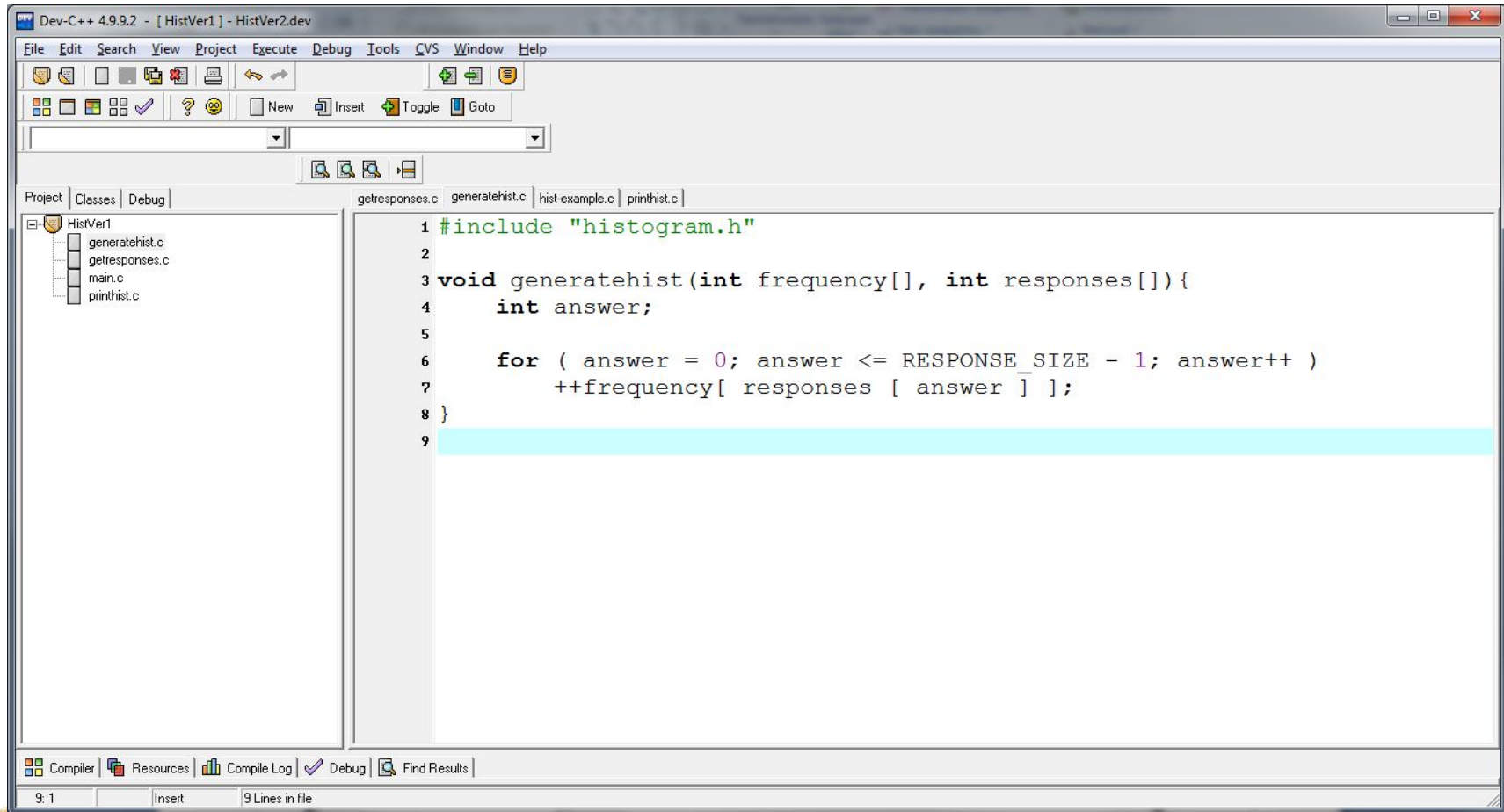
```
Dev-C++ 4.9.9.2 - [HistVer1] - HistVer1.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug | histogram.h | main.c | getresponses.c
HistVer1
  generatehist.c
  getresponses.c
  main.c
  printhist.c
1 #include "histogram.h"
2
3 void getresponses(int data[]) {
4     int i;
5
6     printf("getresponses\n");
7
8
9     for (i=0; i<RESPONSE_SIZE; i++)
10        data[i] = i;
11
12 }
13
6: 1 Insert 13 Lines in file
```

Παράδειγμα

- Άνθρωπος -> γνώμη
- 0 -> 2
- 1 -> 3
- 2 -> 2
- 3 -> 4
- ...
- Πώς διατυπώνεται το πρόβλημα;
- Τι τύπους δεδομένων θα χρησιμοποιήσω;
- Να μετρήσω πόσοι είπαν 0,
 - Πόσοι είπαν 1, πόσοι είπαν 2...
 - Πώς θα το κάνω;



Version 2: γράφουμε τις σωστές συναρτήσεις



The screenshot shows the Dev-C++ IDE with the following code in the editor:

```
1 #include "histogram.h"
2
3 void generatehist(int frequency[], int responses[]){
4     int answer;
5
6     for ( answer = 0; answer <= RESPONSE_SIZE - 1; answer++ )
7         ++frequency[ responses [ answer ] ];
8 }
9
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, Help), a toolbar, a project explorer on the left showing a project named 'HistVer1' with files 'generatehist.c', 'getresponses.c', 'main.c', and 'printhis.c', and a status bar at the bottom indicating '9:1' and '9 Lines in file'.



Πρότυπα συναρτήσεων



```
#include <stdio.h>
#include <stdlib.h>

#define RESPONSE_SIZE 40
#define FREQUENCY_SIZE 11

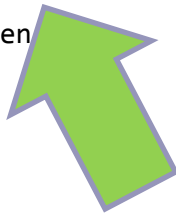
void generatehist(int [], int []);
void printhist(int []);
void printstars(int );

int main()
{
    int frequency[ FREQUENCY_SIZE ] = { 0 };
    int responses[ RESPONSE_SIZE ] =
        { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
          1, 6, 3, 8, 6, 10, 3, 8, 2, 7,
          6, 5, 7, 6, 8, 6, 7, 5, 6, 6,
          5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };

    generatehist(frequency, responses);

    printhist(frequency);

    return 0;
}
```



Υλοποίηση συναρτήσεων



```
void generatehist(int frequency[], int responses[]){
    int answer;

    for ( answer = 0; answer <= RESPONSE_SIZE - 1; answer++ )
        ++frequency[ responses [ answer ] ];
}

void printhist(int frequency[]) {
    int rating;

    printf( "%s%17s Bar\n", "Rating", "Frequency" );
    for ( rating = 1; rating <= FREQUENCY_SIZE - 1; rating++ ) {
        printf( "%6d%17d ", rating, frequency[ rating ] );
        printstars(frequency[rating]);
        printf("\n");
    }
}

void printstars(int len){
    int i;

    for (i=0;i<len;i++)
        putchar('*');
}
```



Η λεκτική περιγραφή αντιστοιχίζεται σε κλήση συναρτήσεων.


```

#include <stdio.h>
#define RESPONSE_SIZE 40
#define FREQUENCY_SIZE 11

int main()
{
    int answer, rating, frequency[ FREQUENCY_SIZE ] = { 0 };
    int responses[ RESPONSE_SIZE ] =
        { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
          1, 6, 3, 8, 6, 10, 3, 8, 2, 7,
          6, 5, 7, 6, 8, 6, 7, 5, 6, 6,
          5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };

    for ( answer = 0; answer <= RESPONSE_SIZE - 1; answer++ )
        ++frequency[ responses [ answer ] ];

    printf( "%s%17s\n", "Rating", "Frequency" );

    for ( rating = 1; rating <= FREQUENCY_SIZE - 1; rating++ )
        printf( "%6d%17d\n", rating, frequency[ rating ] );

    return 0;
}

```



Αλφαριθμητικά (strings)

πρόκειται για **πίνακες χαρακτήρων**:

- `char name[30];`
- αρχικοποίηση με
`char name[30] = "abcd";`
το οποίο ισοδυναμεί με
`name[0] = 'a';`
`name[1] = 'b';`
`name[2] = 'c';`
`name[3] = 'd';`
`name[4] = 0 ; /* δηλώνει το τέλος ενός
αλφαριθμητικού */`



Ανάγνωση και εκτύπωση αλφαριθμητικού

- `char str[N_MAX];`
- `scanf ("%s", str);`
- `printf ("%s\n", str);`
- `%s` → αντιστοιχεί σε αλφαριθμητικό
- `str[0]` είναι ο πρώτος χαρακτήρας
- `str` είναι η διεύθυνση του πρώτου στοιχείου
 - `str` είναι το **ίδιο** με `&str[0]`
 - ισχύει για κάθε τύπο πίνακα



Επεξεργασία αλφαριθμητικού ανα χαρακτήρα

```
#include <stdio.h>
int main()
{
    char string1[ 20 ], string2[] = "string literal";
    int i;
    printf(" Enter a string: ");
    scanf( "%s", string1 );
    printf( "string1 is: %s\nstring2: is %s\n"
           "string1 with spaces between characters is:\n",
           string1, string2 );
    for ( i = 0; string1[ i ] != '\0'; i++ )
        printf( "%c ", string1[ i ] );
    printf( "\n" );
    return 0;
}
```



Μια προγραμματιστική τεχνική

- **Εξασφαλίζουμε** ότι μια συνάρτηση μπορεί να αλλάξει τιμές πίνακα **μόνο αν** αναλυτικά το επιτρέψουμε.
- Εφαρμογή της αρχής **ελαχίστου δικαιώματος** (principle of least privilege).



Παράδειγμα

```
#include <stdio.h>
void DisplayName(char []);

int main ( ) {
char astring[10] = "Hello";

DisplayName(astring);
DisplayName(astring);

return 0;
}

void DisplayName(char x[]) {
int i;
for (i=0; x[i]!=0 ; i++)
    printf("%c", x[i]);
x[0]='h';
printf("\n");
}
```



```
#include <stdio.h>
void DisplayName(const char []);

int main ( ) {
char astring[10] = "Hello";

DisplayName(astring);
DisplayName(astring);

return 0;
}

void DisplayName(const char x[]) {
int i;
for (i=0; x[i]!=0 ; i++)
    printf("%c", x[i]);
x[0]='h';
printf("\n");
}
```

Error: assignment of read-only location

Χρήση του προσδιοριστή `const`

- **`const char *c;`**
 - Δείκτης σε σταθερό χαρακτήρα
 - Δεν αλλάζουν τα περιεχόμενα της θέσης στην οποία δείχνει το `c`
- **`char const *c;`**
 - Δείκτης σε σταθερό χαρακτήρα (ίδιο)
- **`char * const p;`**
 - Σταθερός δείκτης σε χαρακτήρα
 - Δεν αλλάζει το `p`
- **`const char * const p;`**
 - Σταθερός δείκτης σε χαρακτήρα
 - Δεν αλλάζει το `p`
 - Δεν αλλάζουν και τα περιεχόμενα της θέσης στην οποία δείχνει το `p`



Αλφαριθμητικό ως πίνακας χαρακτήρων: Παράδειγμα

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    char alphabet[27]; // 26 letters plus trailing zero
    char c;

    for (c='A'; c<='Z'; c++)
        alphabet[c-'A'] = c;

    alphabet[c-'A'] = 0;

    printf("%s", alphabet);

    return 0;
}
```



Δήλωση και ανάθεση τιμής σε αλφαριθμητικό

```
char *strcpy(char *, const char*);
```

```
main ( ) {  
    char name[10];  
    name = "katerina";  
    printf ("%s", name);  
    scanf ("%s", name);  
    printf ("%s", name);  
}
```

Λάθος
(στη C)



```
#include <string.h>  
main ( ) {  
    char name[10];  
    strcpy(name, "katerina");  
    printf ("%s", name);  
    scanf ("%s", name);  
    printf ("%s", name);  
}
```

Σωστός τρόπος:

Σύγκριση αλφαριθμητικών

- Συνάρτηση

```
int strcmp (const char *, const char *)
```

- για σύγκριση αλφαριθμητικών:
 - Επιστρέφει 0 αν είναι ίδια
 - -1 αν το πρώτο όρισμα προηγείται του δεύτερου
 - +1 αν έπεται.
- **Προσοχή!!!** αν θέλω να συγκρίνω δύο αλφαριθμητικά δεν χρησιμοποιώ τον τελεστή ==
 - Αυτός συγκρίνει θέσεις όχι περιεχόμενα!!!



Παράδειγμα σύγκρισης αλφαριθμητικών

```
#include <stdio.h>
#include <stdlib.h>

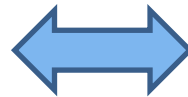
int main(int argc, char *argv[])
{
    char name[100] = "Katerina";
    char user[100] = "";

    printf("Enter your name:");

    while (strcmp(user,name)!=0)
        scanf("%s",user);

    printf("Hello %s\n", user);

    system("PAUSE");
    return 0;
}
```



```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    char name[100] = "Katerina";
    char user[100] = "";

    printf("Enter your name:");

    while (strcmp(user,name))
        scanf("%s",user);

    printf("Hello %s\n", user);

    system("PAUSE");
    return 0;
}
```



Άλλο εννοεί εδώ...

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{

    char name[100] = "Katerina";
    char user[100] = "";

    printf("Enter your name:");
    while (user != "Katerina") {
        scanf("%s",user);
    }

    printf("Hello %s\n", user);

    system("PAUSE");
    return 0;
}
```

- **Δεν κάνει** αυτό που χρειάζεται !!!
- Τι κάνει;
 - Συγκρίνει τη θέση που αρχίζει ο user με τη θέση στην οποία είναι αποθηκευμένο το "Katerina".



Δείκτες (Pointers)

- **Δείκτης**: μεταβλητή στην οποία αποθηκεύουμε **διεύθυνση** θέσης μνήμης.
 - δείχνει **ΠΟΥ** είναι αποθηκευμένα δεδομένα
- **Δήλωση Δείκτη**
<τύπος> *<όνομα δείκτη>;
- **ΠΡΟΣΟΧΗ**: Το όνομα πίνακα **είναι** διεύθυνση, αλλά **δεν είναι** μεταβλητή!!!



Παράδειγμα δήλωσης δείκτη

- `char *ch_ptr;`
- Η μεταβλητή `ch_ptr` περιέχει **διεύθυνση μνήμης** στην οποία είναι αποθηκευμένο δεδομένο τύπου χαρακτήρα.
- `char ch;`
 - Η μεταβλητή `ch` έχει ως αξία χαρακτήρα.
- Μπορούμε να δηλώσουμε δείκτες σε δεδομένα διαφόρων τύπων
 - Βασικών τύπων
 - Κατασκευασμένων τύπων



Παράδειγμα χρήσης δείκτη

```
void main ( ) {  
char ch = 'a', ch2;  
char *ch_ptr ;
```

Δήλωση δείκτη σε χαρακτήρα

```
ch_ptr = &ch ;  
ch2 = *ch_ptr ;
```

```
printf ("%c", ch2);
```

```
}
```

* Περιεχόμενα της θέσης στην
οποία δείχνει ο δείκτης



Πίνακες και δείκτες

- `int arr[10], n ;`
- `*(arr + n) \Leftarrow arr[n]`
- `arr + n \Leftarrow &arr[n]`
- ...αλλά και `n[arr] \Leftarrow *(n+arr) // (!!!)`
- χρησιμοποιούμε δείκτες για να περάσουμε ως όρισμα σε συνάρτηση πίνακες
 - ακριβέστερα: σε ποια διεύθυνση μνήμης βρίσκεται το πρώτο στοιχείο του πίνακα.




```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    char ch, ch2;
    char *ch_ptr ;

    ch = 'a';
    printf( "ch is stored at %X\n", &ch);
    printf( "The value of ch is %c\n", ch);

    //ch_ptr is set to point to ch
    ch_ptr = &ch ;
    printf( "ch_ptr is stored at %X\n", &ch_ptr);
    printf( "The value of ch_ptr is set to %X\n", ch_ptr);

    //contents of ch_ptr (i.e., value of ch)
    //are copied to ch2
    ch2 = *ch_ptr ;
    printf ("ch2 is stored at %X\n", &ch2);
    printf ("value of ch2: %c\n", ch2);

    system("PAUSE");
    return 0;
}

```

```

E:\paliuras\courses\Principles\1213\lecture09\basicchptr\basicchptr.exe
ch is stored at 28FF1E
The value of ch is a
ch_ptr is stored at 28FF18
The value of ch_ptr is set to 28FF1E
ch2 is stored at 28FF1F
value of ch2: a
Press any key to continue . . .

```



Χάρτης μνήμης

Διεύθυνση	Περιεχόμενα μνήμης σε hex	Όνομα μεταβλητής
28FF18	1E	ch_ptr
28FF19	FF	
28FF1A	28	
28FF1B	00	
28FF1C		
28FF1E	61	ch
28FF1F	61	ch2



Σημείωμα αναφοράς

- Copyright Πανεπιστήμιο Πατρών,
Παλιουράς Βασίλειος , Δερματάς Ευάγγελος
«Αρχές Προγραμματισμού ».
Έκδοση: 1.0. Πάτρα 2015
- Διαθέσιμο από τη δικτυακική διεύθυνση
<https://eclass.upatras.gr/modules/>

